

Denne artikel er publiceret i det elektroniske tidsskrift  
**Artikler fra Trafikdage på Aalborg Universitet**  
(Proceedings from the Annual Transport Conference  
at Aalborg University)  
ISSN 1603-9696  
[www.trafikdage.dk/artikelarkiv](http://www.trafikdage.dk/artikelarkiv)



## EN BILLIG GPS DATA ANALYSE PLATFORM

Ove Andersen [xcalibur@cs.aau.dk](mailto:xcalibur@cs.aau.dk) Aalborg Universitet

Nick Christensen [nick@bektra.dk](mailto:nick@bektra.dk) BeKTra

Niels T. Larsen [ntl@bektra.dk](mailto:ntl@bektra.dk) BeKTra

Kristian Torp [torp@cs.aau.dk](mailto:torp@cs.aau.dk) Aalborg Universitet

---

### Abstrakt

Denne artikel præsenterer en komplet software platform til analyse af GPS data. Platformen er bygget udelukkende vha. open-source komponenter. De enkelte komponenter i platformen beskrives i detaljer. Fordele og ulemper ved at bruge open-source diskuteres herunder hvilke IT politiske tiltage, der er nødvendige i en organisation for at bruge open-source. Map-matching diskuteres i flere omgange fordi det har været en udfordring at få implementeret. Platformen er udrullet og taget i produktion hos et mindre selskab. Konklusionen er, at med et meget begrænset budget (15.000,- kr.) kan alle organisationer med et digitalt vejkort og GPS data begynde at lave trafikanalyser på disse data. Det er et krav, at der er passende IT kompetencer tilstede i organisationen.

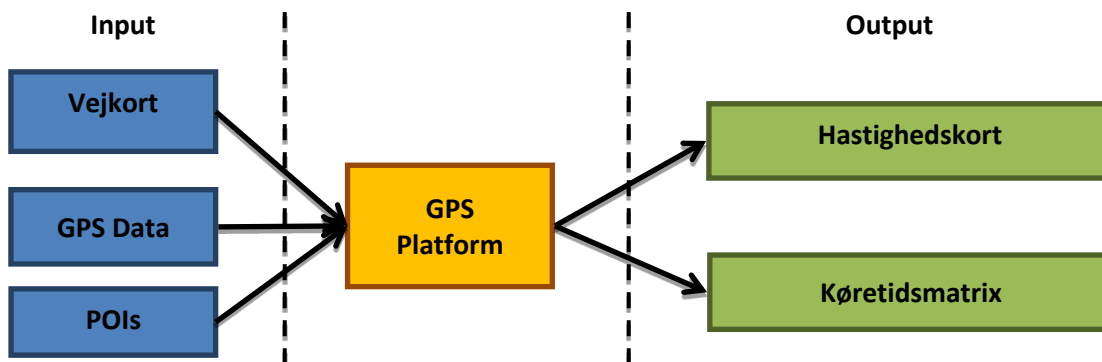
---

### Introduktion

Den artikel beskriver, hvorledes en komplet platform til håndtering og analyse af GPS data kan opbygges af open-source komponenter og anvendes i en organisation som f.eks. et konsulenthus eller en kommune. Platformen er realiseret i et samarbejde mellem et mindre selskab (BeKTra) og et universitet (AAU) og er fra 1. marts 2011 sat i produktion hos BeKTra.

Hovedformålet med platformen er at anvende de store mængder GPS data, der bliver opsamlet (Lahrmann, Agerholm, Tradisauskas, Juhl, & Harms, 2007) (Holm J. , GPS data som grundlag for en national rejsehastighedsdatabase, 2009) til trafikanalyser af f.eks. trængsel (Nielsen, 2003) (COWI A/S, 2004), køretider (Belliss, 2004) (Holm & Foller, 2008) (Torp & Lahrmann, 2009) (Holm J. , GPS realtidspilotprojekt – brug af GPS data til aktuel trafikinformation, 2010) (Hudson & Rhys-Tyler, 2004) og "sorte pletter" på vejnettet (Svendsen, Tradisauskas, & Lahrmann, 2008). Formålet med at skabe en billig platform er at gøre det muligt for mindre organisationer, der har et digitalt vejkort og egen GPS data, at bruge disse data i trafikanalyser.

I BeKTra's tilfælde er ønsket at få beregnet dels rejsehastigheder i vejnettet generelt og dels køretider mellem et veldefineret sæt punkter. Dette er vist i **Figur 1**. Input til platformen er et digitalt vejkort, GPS data samt det sæt af veldefinerede punkter, der skal beregnes køretider i mellem (også kaldet POI = Point Of Interest). Disse input renses (for velkendte fejl og mangler) og gemmes i GPS platformen. Herefter kan der laves analyser, der producerer output hhv. et kort med rejsehastigheder og en såkaldt køretidsmatrix, der indeholder alle beregnede køretider mellem POIerne.



Figur 1 Input og output i BeKTra's brug af GPS platformen

Inputtet til GPS platformen i **Figur 1** kan være et hvilket som helst digitalt vejkort f.eks. over Aarhus, København eller hele Europa. GPS data skal selvfølgelig være tilpasset kortet. Data gemmes i et meget fleksibelt format kaldet et data warehouse (Kimball & Ross, 2002), der kan bruges til praktisk talt alle typer af trafikrelaterede forespørgsler f.eks. trængsel, køretider, mest brugte ruter og "store pletter".

Artiklen beskriver grundene til at anvende open-source komponenter og erfaringerne med disse. Artiklen er ikke en hetz imod betalingssoftware, men en undersøgelse af hvilke open-source komponenter, der er tilgængelige og hvordan disse komponenter kan konfigureres således, at de tilsammen kan danne en platform, der kan anvendes til at rensse, håndtere og forespørge på store mængder GPS data. Et væsentligt punkt i denne undersøgelse er at afdække, hvor mange programmer en organisation som BeKTra er nødsaget til selv at udvikle. Open-source anvendelsen er også direkte et ønske fra BeKTra for at sikre lavere opstartsomkostninger for organisationer, som ønsker at arbejde med udvikling af softwareløsninger indenfor GPS området.

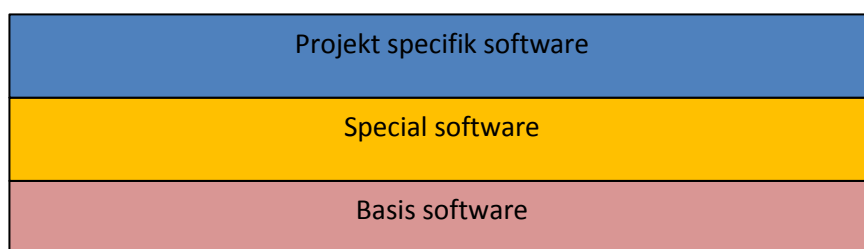
Artiklen er opbygget som følger. I næste afsnit vil først den overordnede platform blive præsenteret. Herefter vil de anvendte open-source komponenter blive beskrevet i detaljer. Så kommer afsnittet om resultater, hardwarekrav samt, hvor hurtigt og hvor meget GPS data, der kan håndteres. I diskussionsafsnittet vil modenheten af platformen og dets komponenter blive vurderet. Herefter drøftes de IT kompetencer, der skal være tilstede i en organisation, der anvender platformen. Slutteligt listes relateret arbejde og en konklusion.

## Metode

I dette afsnit beskrives platformen første generelt. Herefter vil de enkelte komponenter blive beskrevet i yderligere detaljer.

## Software arkitektur

Platformen er realiseret vha. software, der kan placeres i tre lag, som det er vist i **Figur 2** nedenfor. Figuren skal læses sådan, at *basis software* er det mest fundamentale, som *special software* benytter. For at binde platformen sammen er der udviklet *projekt specifik software*, som er det øverste lag.

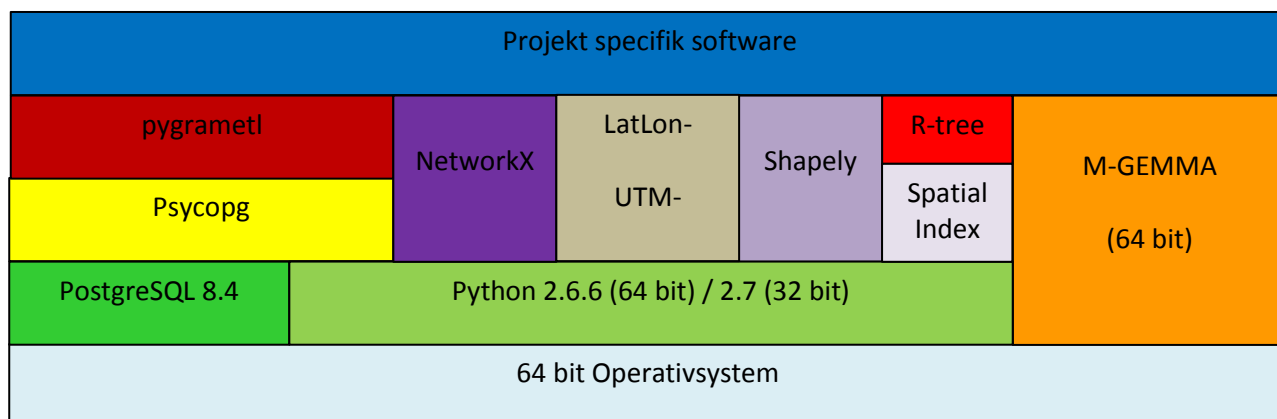


Figur 2 Lagdelt software arkitektur

Som omtalt i introduktionen er det et mål at bruge open-source komponenter for både *basis software* og *special software*. Et naturligt delmål er ligeledes at holde mængden af *projekt specifik software* så lavt som muligt. Dette delmål er nødvendigt fordi denne software skal udvikles i projektet, hvor bemanningen er lav (der er brugt ca. 7 mandemåneder i projektet). En anden grund til at minimere *projekt specifik software* er, at hvis platformen skal anvendes til andre formål, er det her, hovedparten af ændringerne skal ske. En software arkitektur som vist i **Figur 2** er en klassisk løsning i at bygge en platform og kaldes populært en *lagdelt arkitektur*. I det næste afsnit vil indholdet af hvert enkelt lag blive beskrevet i detaljer.

## Open-source komponenter

Dette afsnit beskriver i detaljer de open-source komponenter, der er anvendt i platformen fra det laveste lag i arkitekturen *basis software*, hernæst *special software* og slutteligt *projekt specifik software*.



Figur 3 Detaljeret software arkitektur

Den detaljerede software arkitektur er vist i **Figur 3**. Som det kan ses er den faktiske lagdeling ikke helt så stringent som vist i **Figur 2**. Det forklares senere, hvorfor den stringente opdeling ikke er fastholdt.

## Basis software

Platformen benytter sig af en række basis softwarekomponenter. I softwarearkitekturen ovenover er dette et operativsystem, et database management system (DBMS) og et programmeringssprog.

Det skal være et 64 bit operativsystem, der anvendes f.eks. Microsoft Windows eller en Linux Debian ([www.debian.org](http://www.debian.org)). Begge operativsystemer er afprøvet og begge virker. Der skal anvendes et 64 bit operativsystem pga. forbruget af hukommelse (der skal bruges mere end 4 GB RAM). Windows er ikke en open-source komponent, men er afprøvet fordi dette er standard i mange organisationer.

Som DBMS er valgt PostgreSQL ([www.postgresql.org](http://www.postgresql.org)), som er et meget udbredt DBMS. I DBMSen gemmes alt data. Det er også her, at alle analyser af data udføres. I projektet er anvendt version 8.4 fordi dette var det nyeste, da projektet startede. MySQL DBMSen ([www.mysql.org](http://www.mysql.org)) er uden held blevet afprøvet som alternativ. MySQL kan ikke anvendes, fordi den ikke kan håndtere geo-data (spatielle data) korrekt, dette er en anerkendt og veldokumenteret fejl (mysql.org, 2011).

Som programmeringssprog er valgt Python ([www.python.org](http://www.python.org)). Dette er et meget udbredt og modent programmeringssprog. Som alternative programmeringssprog kunne PHP, Perl eller Java vælges. Disse sprog kan alle anvendes, men er fravalgt fordi deltagerne i projektet har størst forhåndskendskab til Python. Bemærk at flere forskellige versioner af Python er brugt pga. velkendte problemer med komponenter i *special software* laget. Versionen af Python afhænger af om operativsystemet er Windows eller Debian Linux.

## Special software

Når basis softwaren er valgt har dette direkte betydning for hvilke komponenter, der kan vælges i *special software* laget fordi DBMSen og programmeringssproget skal kunne fungere med disse komponenter. For at kunne foretage den såkaldte Extract-Transform-Load (ETL) proces (Kimball & Ross, 2002) er komponenten *pygrame* valgt ([www.pygrame.com](http://www.pygrame.com)). Denne ETL komponent er valgt, fordi det er en af de mest anvendte ETL komponenter lavet i sproget Python. *Pygrame* skal kunne kommunikere med DBMSen og til dette anvendes komponenten *Psycog* ([www.initd.org](http://www.initd.org)). Denne komponent er valgt fordi den er anbefalet af *pygrame*.

De næste special komponenter er *Shapely*, *R-Tree* og *Spatial Index* (alle tilgængelige via [www.gispython.org](http://www.gispython.org)). Disse tre komponenter anvendes til at lave geo-forespørgsler så som, hvilket vejsegment er tættest på et GPS punkt. Alternativer for disse komponenter har ikke været undersøgt, da de valgte komponenter har fungeret godt i projektet, er veldokumenterede og har et større antal udviklere tilknyttet.

Special komponenten *NetworkX* ([networkx.lanl.gov](http://networkx.lanl.gov)) er brugt til at håndtere alle forespørgsler, hvor vejnettet er opfattet som en graf (med knuder og kanter). *NetworkX* komponenten er ryggraden i beregningen af køretidsmatrixen som vist i **Figur 1** og har fungeret upåklageligt i projektet. Komponentens er valgt fordi den er veldokumenteret og har fået gode anmeldelser. Herudover er komponentens hjemmeside tilknyttet en anerkendt amerikansk forskningsinstitution.

Komponenten *LatLonUTMConversion* ([www.pygps.org](http://www.pygps.org)) er anvendt til at konvertere GPS punkter mellem forskellige spatielle reference systemer, specifikt mellem længde-/breddegrader (SRID=4326) og UTM koordinater (SRID=32732). Der er en række alternative komponenter. Den valgte komponent har fungeret uden problemer i projektet.

Den sidste special komponent er *M-GEMMA*, som er en map-matching komponent (Tradisauskas, Juhl, Lahrman, & Jensen, 2007). Det har været klart den største udfordring i projektet at finde en god open-source map-matching komponent. Det er ikke lykkedes at finde en komponent skrevet i programmeringssproget Python. Af denne grund er *M-GEMMA* valgt, selvom den er implementeret i programmeringssproget C++. Python kan bruge C++ kode, når det bygges software, der eksplicit tager sig af kommunikation mellem C++ og Python. Overordnet har *M-GEMMA* fungeret godt, men komponenten har været mere arbejdskraft krævende at få integreret i platformen. Dette diskuteres i detaljer senere i artiklen.

## Projekt specifik software

Som beskrevet i de to foregående sektioner er det lykkedes at finde open-source komponenter for den *basis software* og *special software*, som det er nødvendigt at bruge for at kunne implementere en GPS data analyse platform. Dette betyder, at laget *projekt specifikke software* er holdt til et minimum.

Den projekt specifikke software består hovedsagligt af følgende dele.

- Load af vejkort data. Dette er en triviel opgave, hvis Shapefile formatet anvendes.
- Load af GPS data. Dette er en kompleks opgave fordi den indeholder konfigurationen af en såkaldt ETL proces, hvor data skal renses. ETL processen skal tilpasses hvert enkelt GPS format og kilde. Der er udviklet software så nye GPS formater forholdsvis lette kan tilføjes.
- Udtræk af data til forskellige outputs. Dette er hovedsagligt SQL scripts.
- 

Overordnet er softwaren til ETL processen den eneste komponent, der er special software tilpasset netop det GPS data format, som BeKTra anvender. Skal der anvendes GPS data fra andre kilder, som bruger andre GPS data formater, skal der laves tilføjelser til ETL processen. Det estimeres, at der vil skulle anvendes 1-2 mandemåneder pr. nyt GPS data format. Dette estimat er stærkt afhængigt af, hvor pålidelige GPS kilderne

er. Som et eksempel bliver ETL processen meget mere kompliceret, hvis GPS data er fra kilder, hvor bilisten bevidst har forsøgt at snyde GPS modtageren og der skal tages højde herfor.

## Resultater

Alle resultater, der præsenteres i dette afsnit, er afviklet på en high-end desktop PC Dell Studio XPS 435MT PC med de følgende specifikationer.

- CPU: Intel Core I7-920.
- Hukommelse: 24 GB RAM (6 x 4GB) DDR3-1333.
- Operativsystem: Windows Server 2008 SP2 x64.
- Harddiske: 2 x Western Digital WD6400AAKS, 640GB, 7200 rpm disks i en RAID0 konfiguration med brug af en Intel Matrix Storage Manager, ICH10R.

GPS data er modtaget i tre sendinger og for at test platformen er data indsat i systemet i disse tre sendinger. Det antages, at disse sendinger svare godt til, hvorledes platformen bruges i praksis f.eks. at GPS data opdateres en gang om måneden. Karakteristika for de tre sendinger er vist i tabellen nedenfor.

Sending	Filestørrelse	Rækker	Rækker til punkter		Rækker til ture	
1	17,4 GB	136.045.510	86.232.423	63,4 %	4.890.050	3,6 %
2	27,2 GB	212.413.408	137.480.034	64,7 %	2.380.752	1,1 %
3	6,0 GB	46.536.876	29.974.495	64,4 %	464.957	1,0 %

Tabel 1: GPS data for alle sendinger

Der er i alt cirka 50 GB rå data og cirka 400 millioner rækker. De to sidste koloner viser om data kan bruges som et punkt eller om et køretøj har sendt GPS så ofte, at M-GEMMA map-matching komponent er i stand til at finde den rute et køretøj har taget. Cirka 35 % af al GPS data er sorteret fra. Dette skyldes, at ikke alle køretøjer kan indrapportere deres hastighed. Bemærk yderligere at kun nogle få procent af data kan anvendes til ture. Dette skyldes, at der kan være op til 120 sekunder mellem hver GPS måling fra et køretøj.

Med så store tidsrum mellem to sammenhængende GPS målinger kan M-GEMMA map-matching komponent ikke bestemme en rute. Dette er ikke en fejl i M-GEMMA, men et generelt problem. Det er afprøvet, at M-GEMMA virker fremragende, hvis GPS målinger modtages hvert sekund fra et køretøj. Tidsforbruget for at indsætte GPS målingerne er vist i tabellen nedenfor. *ETL* er hvor lang tid det tager at få rå data ind i databasen og rense data f.eks. at tage alle GPS målinger væk for køretøjer, der ikke kan rapportere hastigheder. *Punkt/tur* er hvor lang tid, der anvendes på at undersøge om en GPS måling kan anvendes i en tur eller kun som et punkt. Dette inkluderer tiden for at map-matche alle GPS målingerne. *Hastighedskort* er tiden for at beregne hastighedskortet, som er et af de to output som vist i **Figur 1**. *Køretidsmatrix* er den tid, det tager at lave køretidsmatrixerne, det andet output vist i **Figur 1**.

Sending	ETL	Punkt/Tur	Hastighedskort	Køretidsmatrix	Total
1	3:05:18	3:39:56	0:04:03	9:49:31	16:38:48
2	5:44:00	3:57:55	0:07:29	9:51:29	19:40:53
3	2:48:18	4:20:40	0:08:12	9:52:53	18:10:03

Tabel 2: Tidsforbrug i timer:minuter:sekunder

Det tager totalt set mellem 17 og 20 timer at indsætte data. Bemærk at her er de cirka 10 timer til at beregne køretidsmatrixerne. Dette tidsforbrug er næsten uafhængigt af datastørrelsen, fordi det er at beregne køretiderne mellem 8884 \* 8884 POIs (cirka 79 millioner beregninger). Tiden for at lave ETL er naturligvis afhængig af antal af rækker i sendingen. Derfor er den største sending 2 den langsomste og den mindste sending 3 den hurtigste. Bemærk at tiden for at skabe et hastighedskort er under 10 minutter.

Samlet set er der muligt på en forholdsvis billig desktop PC at håndtere non-trivielle mængder af GPS data. Disse data kan opdateres om natten. Det er således muligt f.eks. en gang om måneden at opdatere systemet med yderligere GPS data. Det er effektivt at forespørge på data gemt i platformen. Dette kan ses i form af, at et hastighedskort kan skabes på under 10 minutter. Det skal dog bemærkes, at ønskes der mange beregninger, så som det at skabe en køretidsmatrix med cirka 79 millioner beregninger så tager det et betydeligt stykke tid.

## Udrulning

GPS platformen er udviklet på Aalborg Universitet og testet her. Efterfølgende er platformen udrullet hos BeKTra for at komme i produktion. Det har taget 3 dage at installere og træne BeKTra i at anvende platformen. Dette inkluderer tiden til at loadet et digitalt vejkort over hele Danmark samt loadet al GPS data. Yderligere er det afprøvet om MySQL kan anvendes som DBMS i stedet for PostgreSQL. I disse 3 dage har en medarbejder fra Aalborg Universitet siddet hos BeKTra.

## Diskussion

I dette afsnit beskrives først erfaringerne med de open-source komponenter, der er anvendt. Specielt er der fokus på map-matching komponenten. Herefter er der en vurdering og en diskussion af de organisatoriske udfordringer, der er med at anvende en open-source arkitektur.

## Vurdering af open-source komponenterne

Generelt har der været meget få problemer med de open-source komponenter, der er anvendt. Det vurderes, at de komponenter, der er i *basis software* laget: Debian, PostgreSQL og Python alle har en modenhed, dokumentation og udbredelse, så de kan bruges af alle organisationer.

Komponenterne i *special software* laget har en betydelig mindre udbredelse end f.eks. PostgreSQL og Python, men igen har der ikke været problemer med komponenterne i brug. Yderligere er komponenterne generelt veldokumenterede. Det skal bemærkes, at det er tidskrævende at finde og vurdere om en special komponent har den funktionalitet, der ønskes. Herudover kan det være vanskeligt at vurdere om komponenten aktivt bliver vedligeholdt og har en brugerskare, der kan konsulteres for hjælp, hvis der er forståelsesmæssige problemer.

## Map-matching komponenten

Et konkret problem i projektet har været at finde en map-matching komponent. Dette er en meget speciel og avanceret komponent. I projektet er det kun lykkedes at finde én open-source komponent M-GEMMA. Da projektet har valgt at anvende programmeringssproget Python og M-GEMMA er implementeret i C++ har det krævet, at der skabes projekt specifik software til kommunikationen mellem Python og C++. I praksis har det vist sig, at Python og C++ er meget lettet at få til at kommunikere.

En anden problemstilling med M-GEMMA er, at den kræver et meget specifikt input format kaldet NMEA (National Marine Electronics Association). Dette er et proprietært format. Det er undersøgt om dette kunne ændres. En kort gennemgang af kildekode viste, at dette vil være en betydelig arbejdsopgave. Det proprietære format NMEA er blevet reverse-engineered (dvs. genskabt af uafhængige kilder), således det kan anvendes i open-source. Det er derfor valgt at anvende M-GEMMA med NMEA formatet. For at undgå for mange afhængigheder mellem M-GEMMA og resten af komponenterne er det besluttet at afvikle M-GEMMA i en separat proces. Dette har den effekt, at den lagdelte arkitektur som vist i **Figur 2** ikke kan opretholdes og at M-GEMMA håndteres specielt, som vist i den detaljerede software arkitektur i **Figur 3**. Map-matching komponenten kan med rette kaldes software arkitekturens svageste punkt og illustrerer problemet med at anvende open-source software til meget specialiserede opgaver. I arkitekturen er map-matching komponenten derfor isoleret, således den kan udskiftes med f.eks. en kommerciel komponent. Denne isolering har krævet ekstra software. Det skal dog understreges, at M-GEMMA generelt klarer map-matching opgaven godt, hvis datagrundlaget er godt.

## IT kompetencer

Den udviklede platform stiller det krav til organisationer, der anvender at de har professionel IT hjælp enten i form af en IT afdeling eller ekstern konsulentbistand. Det er klart vurderingen, at de nødvendige IT kompetencer ligger på datalog eller softwareingeniør niveau. De nødvendige hovedkompetencer er følgende.

- Installation og konfiguration af større software komponenter som f.eks. PostgreSQL.
- Programmering i sprogene Python, C++ og SQL.
  - Herunder udvikling af programmer der kører parallelt/trådet.
- Design og implementation af data warehouses og ETL flows.

Disse kompetencer er realistiske at finde i en enkelt person med en programmeringsmæssig og data håndteringsmæssig baggrund.

## IT politiske udfordringer

Software platformen kan baseres udelukkende på open-source komponenter. Det skal derfor være tilladt at anvende open-source i organisationen. Visse organisationer er skeptiske over for dette, da open-source komponenter ikke nødvendigvis har en leverandør, der kan kontaktes, hvis der er problemer.

Softwarearkitekturen anvender komponenter, der er kritiske for en organisation sikkerhedsmæssigt. Her tænkes specielt på komponenterne i *basis software* laget. En del organisationer anvender således udelukkende et operativsystem, typisk Microsoft Windows. Det kan være anset for at være for dyrt at skulle understøtte et nyt open-source operativsystem pga. sikkerhedsgodkendelse og manglende kompetencer i IT afdelingen. Af den årsag har projektet testet, at operativsystemet kan være enten Linux (Debian) eller Microsoft Windows.

En anden kritisk komponent i platformen er PostgreSQL DBMSen, hvor alt data gemmes. Som for operativsystemet vælger mange organisationer kun at understøtte et DBMS. Dette gør at f.eks. backup og opgradering er simple. I projektet er det testet, at MySQL *ikke* kan anvendes pga. manglende funktionalitet. Det vurderes, at et kommercielt DBMS fra f.eks. IBM, Microsoft eller Oracle forholdsvis let kan erstatte PostgreSQL. Dog skal det bemærkes, at DBMSen skal understøtte geo-data (spatielle data). Dette er tilkøbsoptioner for visse kommercielle DBMSer eller findes kun i specielle entreprise versioner.

## Datagrundlaget

Alle beregninger af hastighedskort og køretidsmatricer er baseret på GPS data. Hvis disse grunddata ikke er gode er beregningerne baseret på disse data naturligvis heller ikke gode. Det er overraskende, at cirka 35 % af data ikke kan anvendes fordi køretøjerne ikke rapporterer deres hastighed. Det har i løsningen været nødvendigt at identificere, hvilke køretøjer dette drejer sig om og eliminere al data fra disse køretøjer i beregningerne. BeKTra vil fremadrette sikre, at alle køretøjer indrapporterer hastigheden. Dette vil på sigt kraftigt reducere mængden af GPS data, der *ikke* kan anvendes.

I løsningen har der været et fokus på map-matching. Fordi mange køretøjer indrapportere deres position med 30, 60 eller 120 sekunders intervaller kan map-matching komponenten ikke beregne ruterne som køretøjerne har taget. Nærmere undersøgelser har vist, at ruterne generelt ikke kan beregnes af M-GEMMA, hvis intervallet mellem to sammenhængende målinger for ét køretøj er over cirka 10 sekunder. Det skal understreges, at dette ikke er et problem med M-GEMMA, men et general problem fordi, hvis intervallet mellem to målinger bliver for langt er der for mange mulige ruter at vælge imellem specielt i byerne. BeKTra vil fremadrettet forkorte intervallet mellem indrapportering. Hermed vil flere GPS målinger kunne anvendes som ture, hvilket antages kan bidrage til at forbedre nøjagtighed af de beregnede køretider.

Samlet kan det siges, at hvis GPS data skal anvendes til trafikrelaterede information er det afgørende at definere hvilke informationer der skal modtages og hvor ofte de indrapporteres.

## Udrulning

En medarbejder fra AAU har assisteret BeKTra i 3 dage med at udrulle platformen i BeKTra's organisation. Det vurderes at have været for kort tid. Det estimeres at være nødvendigt med en overdragelsesperiode på mellem 7 og 10 arbejdsdage. Dette skyldes til dels, at der mellem AAU og BeKTra har været forskellige opfattelser af, hvad en simpel brugergrænseflade er. AAU har været mere til programmering/konfiguration, hvor BeKTra har været interesseret i en grafisk brugergrænseflade. Der har i projektet ikke været fokus på grafiske brugergrænseflader rettet mod administrative brugere.

At der ikke har været fokus på brugergrænsefladen skyldes, at målet med projektet har været at skabe outputtet i **Figur 1** altså et nyt data grundlag BeKTra kan anvende i deres organisation (hastighedskortet og køretidsmatricerne). Målet har *ikke* været at lave et brugervenligt program altså selve GPS platformen i **Figur 1**. Dette blev til dels et mål i løbet af udrulningen af GPS platformen hos BeKTra. Havde dette mål været klart fra projektet begyndelse kunne en grafisk brugergrænseflade bedre havde været tænkt ind. Dette vil dog forlænge projektet betydeligt, fordi det at skabe en grafisk brugergrænseflade er en større udviklingsopgave. Det vurderes, at tage yderligere 5-7 mandemåneder ekstra at skabe en grafisk brugergrænseflade.

## Udvidelser til software arkitekturen

Ingen software løsning er fuldendt. For softwarearkitekturen præsenteret i denne artikel er der to åbenlyse udvidelsesmuligheder: Visualisering af GPS data og håndtering af rasterdata.

I projektet er der udelukkende fokuseret på at lave output i form af statiske filer hhv. et hastighedskort over hele Danmark og køretidsmatricer. Der er ikke fokuseret på at præsentere data visuelt f.eks. vha. interaktive kort hvor der kan zoomes og udvælges områder til nærmere analyse. Der er en række open-source produkter, der understøtter dette f.eks. uDig ([udig.refractions.net](http://udig.refractions.net)) der har god integration med PostgreSQL. Det vurderes derfor at være forholdsvis let at opnå en visuel præsentation af GPS data.

Rasterdata anvendes i mange sammenhænge bl.a. præsenteres vejrdata vha. rasterdata. Det kunne være relevant at sammenkæde GPS data med vejrdata, men her er en klar begrænsning ved at anvende open-source, da PostgreSQL ikke har understøttelse af rasterdata. Denne understøttelse er planlagt, men der er ingen deadline for, hvornår det bliver tilgængeligt. Hvis rasterdata skal understøttes og integreres med GPS data skal enten eksperimentel software anvendes eller der skal bruges flere forskellige open-source DBMSer. Flere DBMSer vil komplicere software arkitekturen betydeligt.

## Relateret arbejde

Der har nationalt være fokuseret på brugen af GPS data til at beregne køretider på det overordnede vejnet i Danmark i nogle år (Holm & Foller, 2008) (Holm J. , GPS data som grundlag for en national rejsehastighedsdatabase, 2009). Der har yderligere været set på svingtider for et kryds (Torp & Lahrmann, 2009) som et eksempel på, hvor GPS data er anvendt til analyse af lokal trafikafvikling. I denne artikel er fokus på at etablere en billig platform, der tillader alle organisationer at analysere på egne eller fælles GPS data.

I et mere internationalt regi er GPS data anvendt til mange formål f.eks. brændstoføkonomi (Frey, Zhang, & Roupail, 2008) opførsel i trafikken (Rogers, Langley, & Wilson, 1999) (Schoenfelder, Axhausen, Antille, & Bierlaire, 2002). Der har også været fokuseret på at skabe køretids databaser (Pfoser, Tryfona, & Voisard, 2006). Sammenlignet med denne artikel er formålet at vurdere om open-source kan anvendes til at skabe en platform, som kan bruges til forskellige typer af trafikanalyse vha. store GPS data mængder.



## Konklusion

Denne artikel har præsenteret en billig open-source baseret platform, der kan anvendes til analyse af GPS data. Med billig menes, at med hardware for mindre end 15.000,- kr. kan en IT-kyndig organisation starte med at analysere deres GPS data.

De forskellige open-source komponenter er præsenteret, og hvor muligt er alternativer forslået. Generelt er der mange open-source komponenter, der kan anvendes og de er veldokumenterede og har virket uden problemer i projektet. Der har været et konkret problem med at finde en map-matching komponent. Denne er en meget specialiseret komponent og der er kun fundet én open-source løsning, hvor det har været nødvendigt at ændre på den grundlæggende arkitektur for at integrere denne komponent.

Det er beskrevet, hvordan platformen er udrullet hos et selskab og hvordan platformen i dag er i produktion. Løsningen er effektiv og kan håndtere betydelige mængder GPS data. Begrænsningerne ved platformen er opridset. Dette er specielt visualisering og understøttelse af rasterdata. Hovedkonklusionen er, at organisationer med de nødvendige IT kompetencer kan starte meget billigt med at få rensset, gemt og analyseret deres GPS data. Det vurderes at tage 3-9 mandemåneder at starte et sådan projekt.

## Tak til

Harry Lahrmann for meget detaljerede kommentarer på udkast til denne artikel. Projektet er en del af Daisy Innovation ([daisy.aau.dk](http://daisy.aau.dk)) støttet af Den Europæiske Fond for Regionaludvikling.

## Referencer

- Belliss, G. (2004). Detailed Speed and Travel Time Surveys using Low Cost GPS Equipment. IPENZ Transportation Group Technical Conference.
- COWI A/S. (2004). *Projekt Trængsel*.
- Frey, H. C., Zhang, K., & Roupail, N. M. (2008). Fuel use and emissions comparisons for alternative routes, time of day, road grade, and vehicles based on in-use measurements. *Environ Sci Technol*, 42, 2483–2489.
- Holm, J. (2009). GPS data som grundlag for en national rejsehastighedsdatabase. Trafikdage på Aalborg Universitet.
- Holm, J. (2010). GPS realtidspilotprojekt – brug af GPS data til aktuel trafikinformation. Trafikdage på Aalborg Universitet.
- Holm, J., & Foller, J. (2008). GPS pilotprojekt. *Trafikdage på Aalborg Universitet*.
- Hudson, M., & Rhys-Tyler, G. (2004). USING GPS DATA TO CALCULATE THE LENGTH AND VARIABILITY OF FREIGHT VEHICLE JOURNEY TIMES ON MOTORWAYS. TRL.
- Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition)*. Wiley.
- Lahrmann, H., Agerholm, N., Tradisauskas, N., Juhl, J., & Harms, L. (2007). Spar på Farten-et forsøg med Intelligent Farartilpasning baseret på incitament (forsikringsrabat). Trafikdage på Aalborg Universitet.
- mysql.org. (2011). *MySQL 5.5 Reference Manual*.
- National Marine Electronics Association. (u.d.). *NMEA 0183 Standard*. Hentede 10. February 2011 fra NMEA: [http://www.nmea.org/content/nmea\\_standards/nmea\\_083\\_v\\_400.asp](http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp)
- Nielsen, O. A. (2003). Analyse af trængsel og hastigheder vha. GPS-data. *Trafikdage på Aalborg Universitet*.
- Pfoser, D., Tryfona, N., & Voisard, A. (2006). Dynamic Travel Time Maps - Enabling Efficient Navigation. *SSDBM*, (s. 369-378).
- Rogers, S., Langley, P., & Wilson, C. (1999). Mining GPS Data to Argument Raod Models. *KDD-99*, (s. 104-113).
- Schoenfelder, S., Axhausen, K., Antille, N., & Bierlaire, M. (2002). Exploring the potentials of automatically collected GPS data for travel behaviour analysis - a Swedish data source. *13*, 155-179.

- Svendsen, M. S., Tradisauskas, N., & Lahrman, H. (2008). Udpegning af potentielle sorte pletter med floating car data. *Trafikdage*.
- Torp, K., & Lahrman, H. (2009). Metode til beregning af køretider, trængsel og forsinkelser i kryds vha. GPS Data. Aalborg: Trafikdage på Aalborg Universitet.
- Tradisauskas, N., Juhl, J., Lahrman, H. L., & Jensen, C. S. (2007). Map Matching Algorithm for the "Spar paa farten" Intelligent Speed Adaptation Project. Trafikdage på Aalborg Universitet.